

# Méthodes de SLAM visuel

Patrick Rives

Projet ARobAS INRIA Sophia-Antipolis Méditerranée

Vendredi 23 Janvier 2009

# Plan de l'exposé

- 1 Généralités sur le SLAM
- 2 Approche "Bayésienne"
- 3 Construire des cartes locales
- 4 Approche "Optimisation"
- 5 Conclusion

# Plan de l'exposé

- 1 Généralités sur le SLAM
- 2 Approche "Bayésienne"
- 3 Construire des cartes locales
- 4 Approche "Optimisation"
- 5 Conclusion

# Plan de l'exposé

- 1 Généralités sur le SLAM
  - Introduction
  - Définition et notations
  - Bearing only Vs. Range and Bearing
- 2 Approche "Bayésienne"
- 3 Construire des cartes locales
- 4 Approche "Optimisation"
- 5 Conclusion

# Localisation et cartographie par SLAM visuel

## Quelques défis du SLAM en robotique mobile :

- cartes de grande taille,
- environnement évolutif et/ou dynamique,
- SLAM avec 6 degrés de liberté,
- garanties sur la position de la carte et des amers,
- intégrer dans un formalisme unique : Slam, navigation autonome.

# Localisation et cartographie par SLAM visuel

## Spécificités du SLAM visuel :

- richesse de l'information visuelle  $\Rightarrow$  association des données facilitée
- problèmes de changements d'éclairément (capteur passif),
- Capteur projectif  $\Rightarrow$  difficulté supplémentaire d'initialisation des amers,
- quantité importante d'information à traiter.

# Localisation et cartographie par SLAM visuel

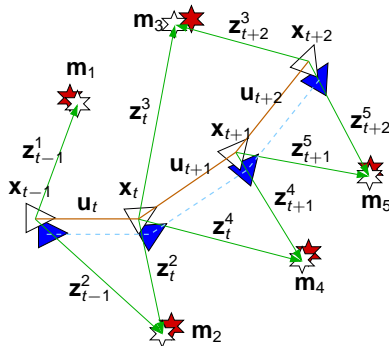
## Contributeurs à cet exposé :

- Approches probabilistes consistantes pour le SLAM (C. Joly),
- Structuration des cartes et représentations (C. Joly),
- Feature-based SLAM en vision omnidirectionnelle (C. Joly, H. Hadj AbdeKader)
- Approches directes en SLAM monoculaire (G. Silveira, E. Malis),

# Définition et notations

Le SLAM est un problème d'inférence.

**But :** obtenir  $\mathbf{x}_t$  (ou toute la trajectoire) et la position des amers  $\mathbf{m}$  à partir de mesures  $\mathbf{z}$  *bruitées*.





# Définition et notations

## Formulation "Bayésienne"

- modèle de perception :  
 $p(\mathbf{z}_t | \mathbf{x}_t, \mathbf{m})$
- modèle d'évolution  
 $p(\mathbf{x}_t | \mathbf{x}_{t-1}, \mathbf{u}_t)$

## Formulation "optimisation"

- modèle de perception  
 $z = h(\mathbf{x}, \mathbf{m})$
- fonction de coût  
 $d(h(\mathbf{x}, \mathbf{m}), \mathbf{z})$

But :

$$(\mu, \Sigma) \text{ tq } p(\mathbf{x}_t, \mathbf{m} | \mathbf{z}_{0:t}, \mathbf{u}_{0:t}, \mathbf{x}_0)$$

But :

$$(\hat{\mathbf{x}}, \hat{\mathbf{m}}) \text{ tq } \min_{\mathbf{x}, \mathbf{m}} \{d(h(\mathbf{x}, \mathbf{m}), \mathbf{z})\}$$

# Bearing only Vs. Range and Bearing

	Avantages	Inconvénients
<i>Range and bearing</i>	<ul style="list-style-type: none"> <li>● Information de profondeur (en plus de la direction) disponible à chaque mesure</li> <li>● Facilite les algorithmes de filtrage lors de la résolution</li> </ul>	<ul style="list-style-type: none"> <li>● Contraintes technologiques. Exemples :               <ul style="list-style-type: none"> <li>● Laser 3D : temps d'acquisition long</li> <li>● Caméras stéréo : contraintes mécaniques, mise en correspondance coûteuse en calcul</li> </ul> </li> <li>● Prix des capteurs élevé</li> </ul>
<i>Bearing - Only</i>	<ul style="list-style-type: none"> <li>● Bon marché</li> <li>● Temps d'acquisition rapide</li> <li>● Facile à monter</li> </ul>	<ul style="list-style-type: none"> <li>● Localisation d'amer incomplète (manque la profondeur)</li> </ul>

# Plan de l'exposé

- 1 Généralités sur le SLAM
- 2 Approche "Bayésienne"**
- 3 Construire des cartes locales
- 4 Approche "Optimisation"
- 5 Conclusion

# Plan de l'exposé

- 1 Généralités sur le SLAM
- 2 Approche "Bayésienne"
  - Exemple du "Bearing Only SLAM"
  - Slam en environnement à grande échelle
  - SAM : Smoothing and Mapping
- 3 Construire des cartes locales
- 4 Approche "Optimisation"
- 5 Conclusion

# Exemple du "Bearing Only SLAM"

Robot

$$\mathbf{x}_t = [x_t \ y_t \ \theta_t]^T$$

Amer(i)

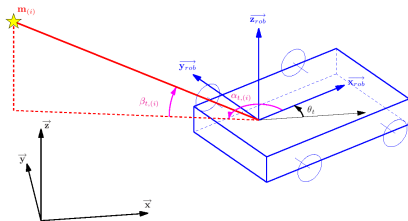
$$\mathbf{m}_{(i)} = [x_{(i)} \ y_{(i)} \ z_{(i)}]^T$$

Commande

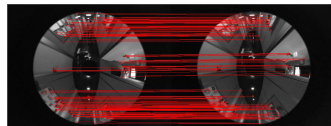
$$\mathbf{u}_t = [v_t \ \omega_t]^T$$

Observations(i)

$$\mathbf{z}_{t,(i)} = [\alpha_{t,(i)} \ \beta_{t,(i)}]^T$$



Modélisation



Mise en correspondance

# Slam en environnement à grande échelle

## Constat

Les méthodes standards de SLAM (EKF) deviennent rapidement inconsistantes dans des environnements à grande échelle.

⇒ Principalement dû à la linéarisation non valide des équations d'évolution et d'observation

## Solutions

- Approche SAM (Smoothing and Mapping)  
 $p(\mathbf{x}_{0:t}, \mathbf{m} | \mathbf{z}_{0:t}, \mathbf{u}_{0:t}, \mathbf{x}_0)$
- Structuration en cartes locales consistantes

# SAM : Smoothing and Mapping

## Principe

- Calculer la densité de probabilité de la **trajectoire** du robot et de la carte :

$$p(\mathbf{x}_{1:t}, \mathbf{m} | \mathbf{x}_0, \mathbf{z}_{0:t}, \mathbf{u}_{1:t}) \text{ au lieu de } p(\mathbf{x}_t, \mathbf{m} | \mathbf{x}_0, \mathbf{z}_{0:t}, \mathbf{u}_{1:t})$$

Toute la trajectoire

Dernière position (EKF)

- Travailler dans l'espace matrice d'information / vecteur d'information ( $\Omega = \Sigma^{-1}$ ,  $\xi = \Sigma^{-1}\mu$ )

# SAM : Smoothing and Mapping

## Comparaison avec l'EKF :

	Consistance	Calculs / Mémoire
EKF	Ne remet pas en cause la trajectoire. Une inconsistance à un instant donné sera propagée pour toute la suite	Quadratique avec le nombre d'amers
SAM	Remet en cause la trajectoire. Permet de limiter les inconsistances	Dimension de l'état très importante mais matrice d'information éparses Temps de calculs comparables



# Plan de l'exposé

- 1 Généralités sur le SLAM
- 2 Approche "Bayésienne"
- 3 Construire des cartes locales**
- 4 Approche "Optimisation"
- 5 Conclusion

# Plan de l'exposé

- 1 Généralités sur le SLAM
- 2 Approche "Bayésienne"
- 3 Construire des cartes locales**
  - Intérêt
  - Quand démarrer une nouvelle carte ?
  - Gestion des cartes locales
  - Résultats
- 4 Approche "Optimisation"
- 5 Conclusion

# Intérêt :

## Objectif : réduire les incertitudes autour du robot

- Améliorer la sécurité des tâches à court terme effectuées par le robot (planification, contrôle).
- Garder un contrôle sur la qualité de l'estimation non linéaire (limiter les problèmes d'inconsistance)

## Solutions envisagées

- 1 Carte robocentrée (à nombre limité d'amers ou non)
  - Cas du SAM : évaluer les **trajectoires** des amers dans le repère du robot  
⇒ dimension trop importante
  - EKF obligatoire : risque d'inconsistance
- 2 **Cartes locales en réinitialisant les incertitudes**
  - SAM utilisable ⇒ meilleure consistance
  - Permet de représenter la trajectoire entière du robot *a posteriori*

# Quand démarrer une nouvelle carte ? (1/2)

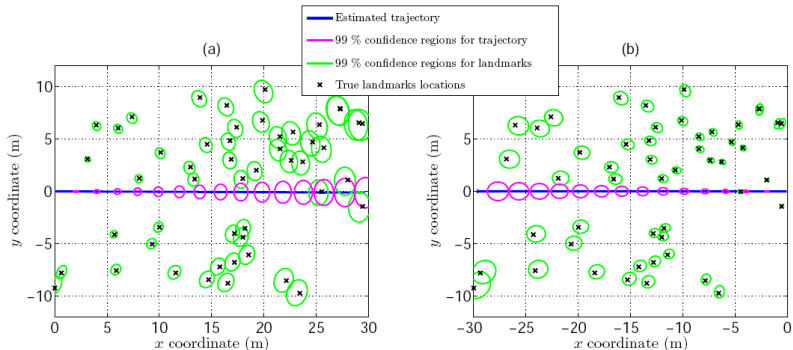
## Données à l'instant $t$ :

- Densité de probabilité *a posteriori* :  $p(\mathbf{x}_{1:t}, \mathbf{m} | \mathbf{x}_0, \mathbf{z}_{0:t}, \mathbf{u}_{1:t})$
- Marginalisation des positions immédiate :  $p(\mathbf{m} | \mathbf{x}_0, \mathbf{z}_{0:t}, \mathbf{u}_{1:t})$
- $\Rightarrow$  matrice de covariances pour chaque amer  $i$  :  $\Sigma^i_{|\mathbf{x}_0}$

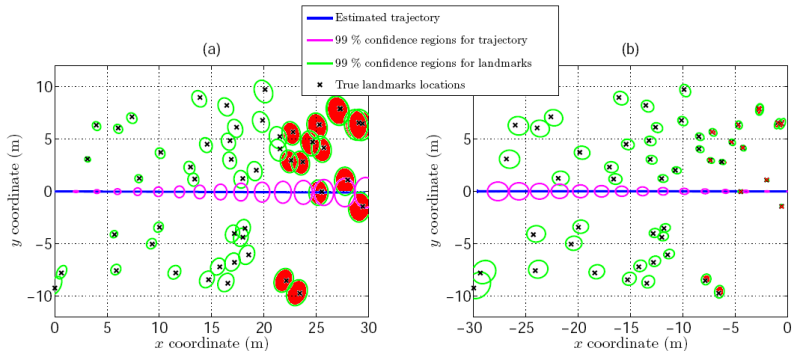
## Paramètres déduits en changeant le point de conditionnement

- Calculer  $p(\mathbf{m} | \mathbf{x}_t, \mathbf{z}_{0:t}, \mathbf{u}_{1:t})$  à partir de  $p(\mathbf{x}_{1:t}, \mathbf{m} | \mathbf{x}_0, \mathbf{z}_{0:t}, \mathbf{u}_{1:t})$
- Solution que l'on aurait obtenue en supposant certaine **la dernière position du robot au lieu de la première**
- $\Rightarrow$  matrice de covariances pour chaque amer  $i$  :  $\Sigma^i_{|\mathbf{x}_t}$

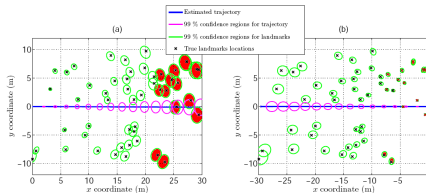
# Quand démarrer une nouvelle carte ? (2/2)



# Quand démarrer une nouvelle carte ? (2/2)



# Quand démarrer une nouvelle carte ? (2/2)



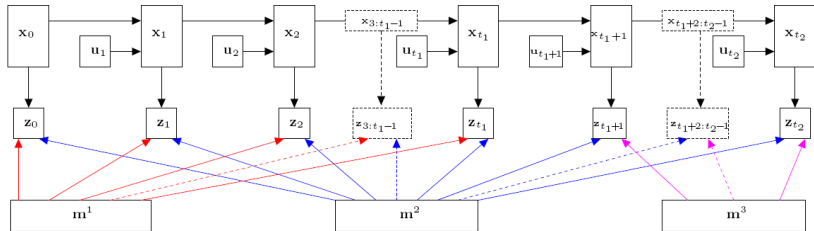
## Critère

- Idée : l'incertitude associée aux derniers amers vus à l'instant  $t$  doit être beaucoup plus faible lorsque l'on suppose connue la position finale du robot que lorsqu'on suppose connue la position initiale
- Calcul pour chacun des derniers amers de  $\sqrt{\frac{\det \Sigma^j | x_t}{\det \Sigma^j | x_0}}$
- Si la moyenne des coefficients est inférieure à un certain seuil : on initialise une nouvelle carte

# Gestion des cartes locales (1/2)

## Données :

- 0 : instant initial
- $t_1$  : fin de la première carte
- $t_2$  : fin de la seconde carte
- $m^1$  : amers uniquement visibles dans la première carte
- $m^2$  : amers communs aux deux cartes
- $m^3$  : amers uniquement visibles dans la seconde carte

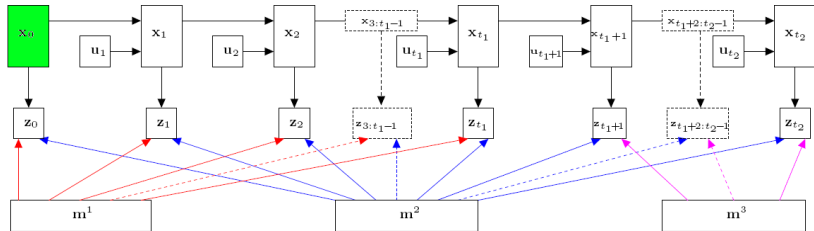




# Gestion des cartes locales (1/2)

## Données :

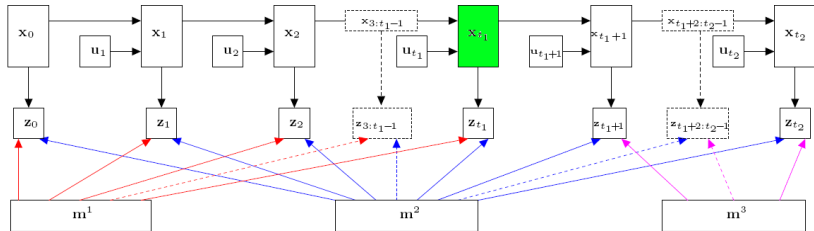
- 0 : instant initial
- $t_1$  : fin de la première carte
- $t_2$  : fin de la seconde carte
- $m^1$  : amers uniquement visibles dans la première carte
- $m^2$  : amers communs aux deux cartes
- $m^3$  : amers uniquement visibles dans la seconde carte



# Gestion des cartes locales (1/2)

## Données :

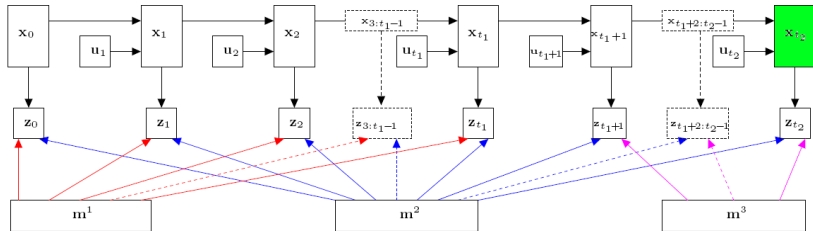
- 0 : instant initial
- $t_1$  : fin de la première carte
- $t_2$  : fin de la seconde carte
- $m^1$  : amers uniquement visibles dans la première carte
- $m^2$  : amers communs aux deux cartes
- $m^3$  : amers uniquement visibles dans la seconde carte



# Gestion des cartes locales (1/2)

## Données :

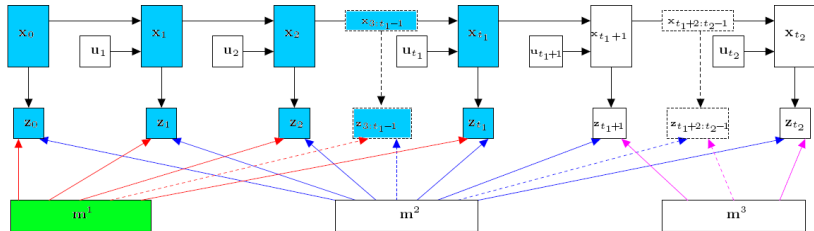
- 0 : instant initial
- $t_1$  : fin de la première carte
- $t_2$  : fin de la seconde carte
- $m^1$  : amers uniquement visibles dans la première carte
- $m^2$  : amers communs aux deux cartes
- $m^3$  : amers uniquement visibles dans la seconde carte



# Gestion des cartes locales (1/2)

## Données :

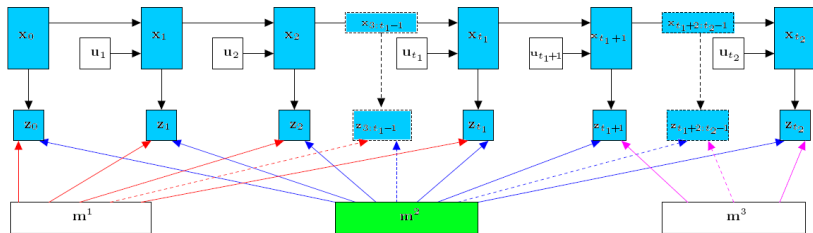
- 0 : instant initial
- $t_1$  : fin de la première carte
- $t_2$  : fin de la seconde carte
- $m^1$  : amers uniquement visibles dans la première carte
- $m^2$  : amers communs aux deux cartes
- $m^3$  : amers uniquement visibles dans la seconde carte



# Gestion des cartes locales (1/2)

## Données :

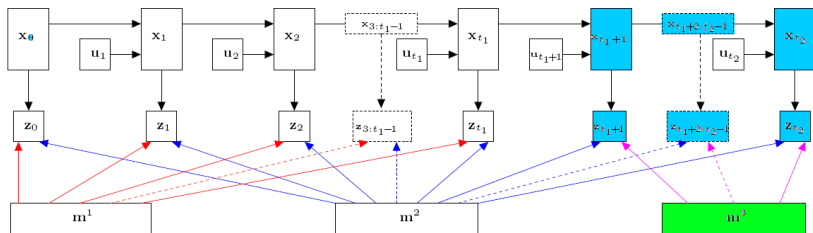
- 0 : instant initial
- $t_1$  : fin de la première carte
- $t_2$  : fin de la seconde carte
- $m^1$  : amers uniquement visibles dans la première carte
- $m^2$  : amers communs aux deux cartes
- $m^3$  : amers uniquement visibles dans la seconde carte



# Gestion des cartes locales (1/2)

## Données :

- 0 : instant initial
- $t_1$  : fin de la première carte
- $t_2$  : fin de la seconde carte
- $m^1$  : amers uniquement visibles dans la première carte
- $m^2$  : amers communs aux deux cartes
- $m^3$  : amers uniquement visibles dans la seconde carte



# Gestion des cartes locales (1/2)

## Données :

- 0 : instant initial
- $t_1$  : fin de la première carte
- $t_2$  : fin de la seconde carte
- $\mathbf{m}^1$  : amers uniquement visibles dans la première carte
- $\mathbf{m}^2$  : amers communs aux deux cartes
- $\mathbf{m}^3$  : amers uniquement visibles dans la seconde carte

## Problème :

Est-il possible de calculer la densité associées aux positions du robot entre  $t_1 + 1$  et  $t_2$  ainsi que les amers  $\mathbf{m}^2$  et  $\mathbf{m}^3$  en prenant en compte l'**intégralité des mesures** ?

$$\implies p(\mathbf{x}_{t_1+1:t_2}, \mathbf{m}^2, \mathbf{m}^3 | \mathbf{x}_{t_1}, \mathbf{z}_{0:t_2}, \mathbf{u}_{1:t_2})$$

# Gestion des cartes locales (2/2)

## Théorème :

$$p(\mathbf{x}_{t_1+1:t_2}, \mathbf{m}^2, \mathbf{m}^3 | \mathbf{x}_{t_1}, \mathbf{z}_{0:t_2}, \mathbf{u}_{0:t_2}) \propto \\ p(\mathbf{x}_{t_1+1:t_2}, \mathbf{m}^2, \mathbf{m}^3 | \mathbf{x}_{t_1}, \mathbf{z}_{t_1+1:t_2}, \mathbf{u}_{t_1+1:t_2}) \times p(\mathbf{m}^2 | \mathbf{z}_{0:t_1}, \mathbf{u}_{0:t_1}, \mathbf{x}_{t_1})$$

## Interprétation

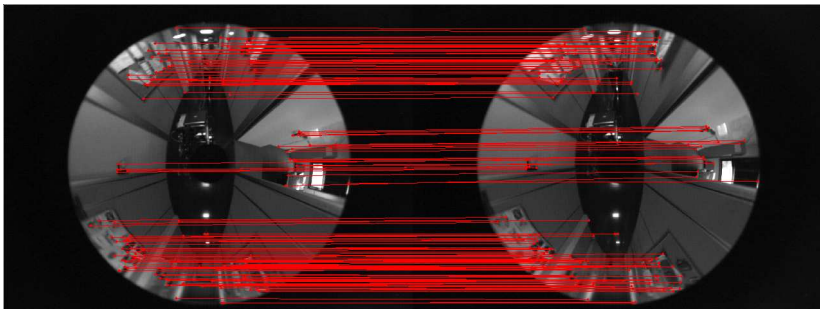
**1er facteur :** s'obtient en appliquant un algorithme de SAM classique n'utilisant que les mesures et commandes de la seconde carte

**2ème facteur :** se déduit de la première carte

**Multiplication :** vecteur et matrice d'observation se déduisent par addition des paramètres déduits des deux facteurs

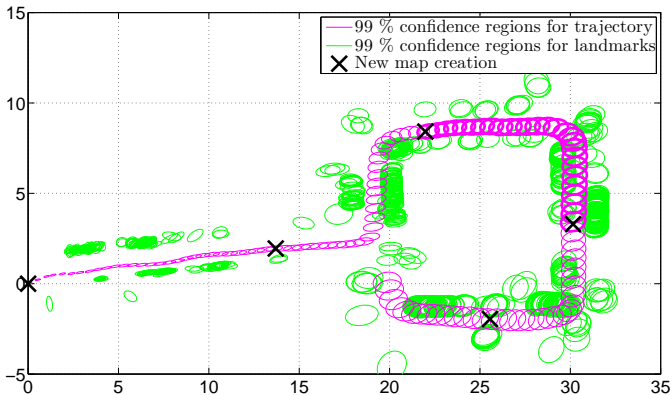


# Résultats : SLAM par vision catadioptrique



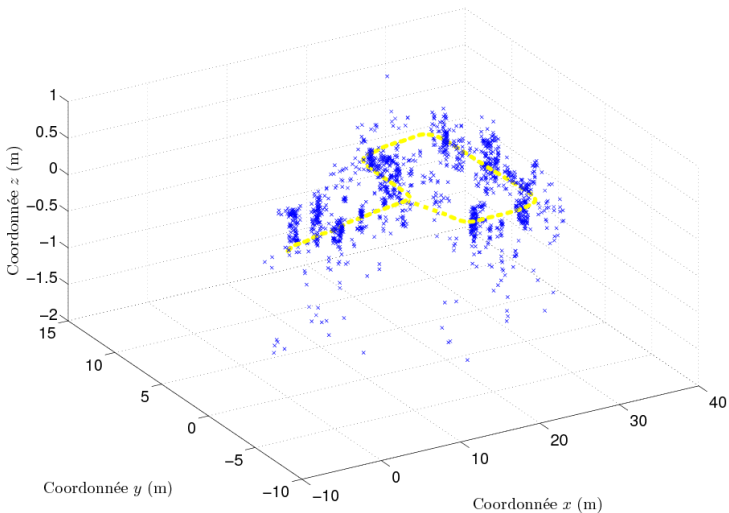
Mise en correspondance

# Résultats : SLAM par vision catadioptrique

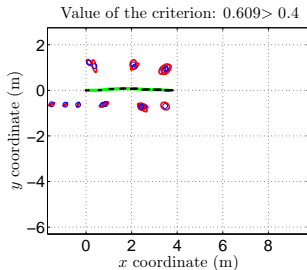
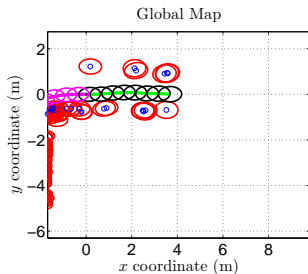


Carte globale

# Résultats : SLAM par vision catadioptrique

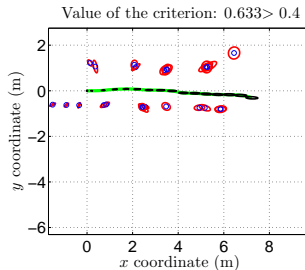
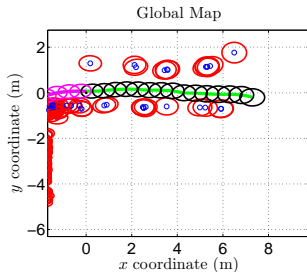


# Résultats : Structuration en cartes locales



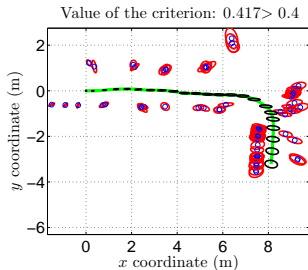
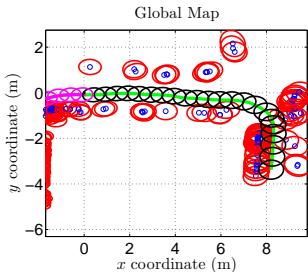
Etape 1

# Résultats : Structuration en cartes locales



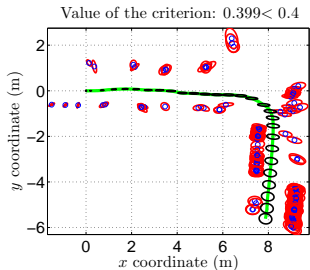
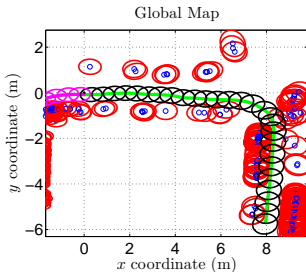
Etape 2

# Résultats : Structuration en cartes locales



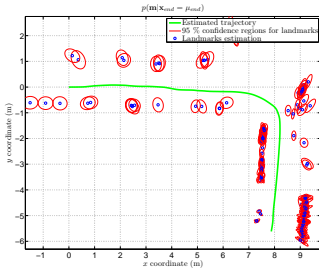
Etape 3

# Résultats : Structuration en cartes locales

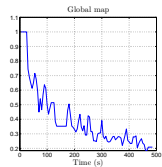
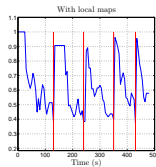


Etape 4

# Résultats : Structuration en cartes locales



Reconditionnement



Critère de décision



# Plan de l'exposé

- 1 Généralités sur le SLAM
- 2 Approche "Bayésienne"
- 3 Construire des cartes locales
- 4 Approche "Optimisation"**
- 5 Conclusion

# Plan de l'exposé

- 1 Généralités sur le SLAM
- 2 Approche "Bayésienne"
- 3 Construire des cartes locales
- 4 Approche "Optimisation"
  - Introduction
  - Modélisation
  - Méthode efficace d'optimisation
  - Algorithme de suivi visuel robuste par ESM
  - Application au SLAM monoculaire
  - Résultats

# Méthodes directes

Trouver  $\tilde{\mathbf{x}}^c = h^{-1}(\mathbf{z}) \in G^c$

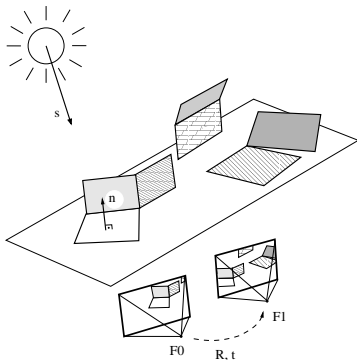
## Solution du problème inverse

- Formulé comme la solution d'un problème d'optimisation :
  - **algorithme d'optimisation efficace**
  - **algorithmes avec un grand domaine de convergence**
- Environnements non rigides et avec éclairage variable :
  - **changements de géométrie (objets déformables)**
  - **changements d'illuminations (spécularités, ombres, ...)**

## Contributions

- modélisation hiérarchique
- méthodes denses en temps réel

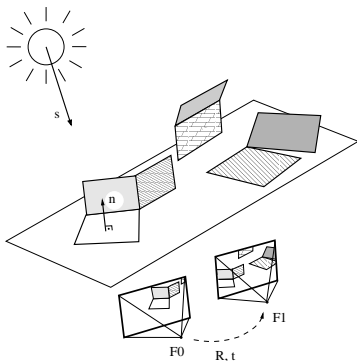
# Méthodes directes



Comment choisir  $w$  ?

$$\min_{R, t, n_1, n_2, \dots, s_1, s_2, \dots} \{I(w(R, t, n_1, n_2, \dots, s_1, s_2, \dots)) - I^*\} ?$$

# Méthodes directes



Comment choisir  $w$  ?

$$\min_{R, t, n_1, n_2, \dots, s_1, s_2, \dots} \{I(w(R, t, n_1, n_2, \dots, s_1, s_2, \dots)) - I^*\} ?$$

# Modélisation

- Modèle générique de la fonction de perception

$$\mathcal{I}'(\mathbf{x}^c, \mathbf{p}^*) = \mathcal{S}(\Gamma, \mathbf{p}^*) \bullet \mathcal{I}(\mathbf{w}(\mathbf{g}^u, \mathbf{p}^*)) + \beta$$

avec  $\mathbf{x}^c = \{\mathbf{g}^c, \mathbf{h}\}$

- Paramètres locaux et globaux pour
  - les aspects géométriques :  $\mathbf{g}^c = \{\mathbf{R}, \mathbf{t}, (\mathbf{z}^*)^{-1} : (\mathbf{z}^*)^{-1} > 0\}$
  - les aspects photométriques :  $\mathbf{h} = \{\Gamma, \beta\}$
- Minimisation récursive de l'erreur.

$$\min_{\tilde{\mathbf{x}}^c \in \mathcal{G}^c} \frac{1}{2} \sum_i [\mathcal{I}'(\tilde{\mathbf{x}}^c \circ \hat{\mathbf{x}}^c, \mathbf{p}_i^*) - \mathcal{I}^*(\mathbf{p}_i^*)]^2$$

# Méthode efficace d'optimisation

- Introduction hiérarchique des paramètres géométriques
  - 1 rotation pure
  - 2 rotation + translation + structure
- Optimisation par *ESM Efficient Second-order Minimization* (S. Benhimane, E. Malis)
  - Ne nécessite que le calcul en ligne du Jacobien de l'image courante
  - Implémentation multi-échelle temps réel





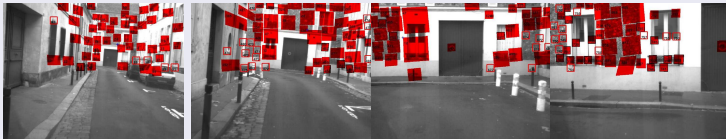
# Application au SLAM monoculaire

## Détection/gestion des régions d'intérêts (Rois)

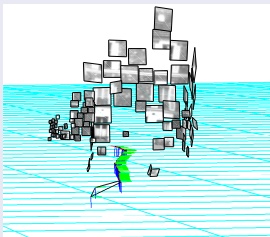
- Nécessaire dans le cas de longs déplacements
- Critère permettant de
  - sélectionner les "bonnes" régions d'intérêt dans l'image (conditionnement du gradient)
  - les insérer dans l'algorithme de Slam (du projectif vers l'Euclidien)
  - rejeter les régions correspondant à des "outliers" (résidu de la minimisation, index photométrique)

# Résultats

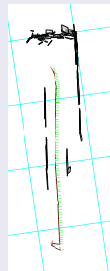
## Séquence Rue droite



Input images



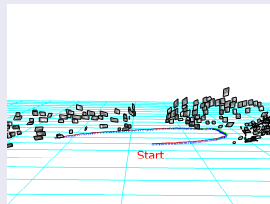
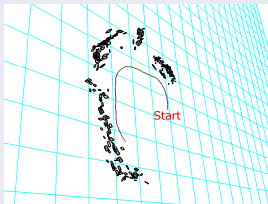
SLAM results



## Séquence Rond-point



Input images



SLAM results

# Plan de l'exposé

- 1 Généralités sur le SLAM
- 2 Approche "Bayésienne"
- 3 Construire des cartes locales
- 4 Approche "Optimisation"
- 5 Conclusion**

# Plan de l'exposé

- 1 Généralités sur le SLAM
- 2 Approche "Bayésienne"
- 3 Construire des cartes locales
- 4 Approche "Optimisation"
- 5 Conclusion**

# Conclusion

## Approche probabiliste

### ● Avantages

- Permet de manipuler des notions de consistance et de qualité,
- Complexité raisonnable dans le cas de structuration en cartes locales,
- Approche de mise en correspondance autorisant des déplacements importants,
- Limitation du drift dans le cas du SAM

### ● Limitations

- Sensibilité des méthodes basées sur des représentations éparses,
- Robustesse des points d'intérêts et de l'association de donnée,
- Modèle 3d peu riche,

# Conclusion

## Approche directe

### ● Avantages

- Utilise toute l'information dans l'image,
- Insertions des Rols dès leurs détections (no delayed initialization)
- Modèle 3d riche  $\Rightarrow$  géométrique + photométrique,
- Robustesse au changement d'illuminations
- Robustesse/précision des approches denses

### ● Limitations

- Difficulté à manipuler des notions de consistance et de qualité
- Complexité algorithmique plus grande,
- Relativement petits déplacements pour assurer la convergence,
- Méthode d'odométrie visuelle  $\Rightarrow$  limite le drift mais ne le supprime pas !